Computational Algorithms and Numerical Dimensions

www.journal-cand.com

Com. Alg. Num. Dim.Vol. 4, No. 1 (2025) 57–64.

Paper Type: Original Article

An Iterative Method Based on Average Quadrature Formula

Tusar Singh[1], Dwiti Krushna Behera[2,*], Shno Othman Ahmed[3], Rostam Kareem Saeed[4]

[1] Department of Mathematics, Siksha O Anusandhan University, Bhubaneswar, Odisha, India; tusarsingh@soa.ac.in.
[2] Department of Mathematics, Ravenshaw University, Cuttack, Odisha, India; dkb_math@ravenshawuniversity.ac.in.
[3] Department of Computer, College of Science, Salahaddin University-Erbil, Erbil, Iraq; shno.ahmed@su.edu.krd.
[4] Department of Mathematics, College of Science, Salahaddin University-Erbil, Kurdistan Region, Erbil, Iraq; rostam.saeed@su.edu.krd.

Citation:

Received: 19 October 2024 *Revised: 20 December 2024* *Accepted: 08 Januray 2025*	Singh, T., Behera, D. K., Othman Ahmed, S., & Saeed, R. K. (2025). An iterative method based on the average quadrature formula. *Computational algorithms and numerical dimensions*, *4*(1), 57-64.

Abstract

In our research, a new third-order iterative method has been introduced. This method involves the creation of a new quadrature formula by averaging Simpson's $\frac{1}{3}$rd and Trapezoidal rules. The newly developed quadrature formula is then used to establish the new iterative scheme, which modifies the Newton-Raphson method. It has been demonstrated that the new iterative technique exhibits a convergence order of 3. Finally, examples have been provided to illustrate the effectiveness of the new process. The results indicate that the new approach finds the root of the nonlinear equation in fewer iterations compared to other methods, suggesting the potential superiority of our newly developed scheme.

Keywords: Trapezoidal rule, Simpson's $\frac{1}{3}$rd, Newton's method, Order of convergence, Iterative method.

1 | Introduction

Iterative methods are powerful mathematical tools that have been refined over centuries, reflecting the evolution of human thought and computational techniques. They play a vital role in advancing computational mathematics, engineering, and the applied sciences.

The Babylonians used early forms of iteration to calculate square roots, employing an averaging method. Archimedes utilized iterative techniques in his approximation method, demonstrating early concepts of convergence. Sir Isaac Newton formulated his process for finding the roots of a function, which became foundational for many iterative techniques. This process is defined by [1].

Corresponding Author: dkb_math@ravenshawuniversity.ac.in
https://doi.org/10.22105/cand.2024.485552.1155

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (1)$$

The quadrature rule serves as a valuable numerical technique for approximating the definite integral of a function. Its application in the development of iterative methods is widely recognized and has proven to be significant. It is worth noting that the use of quadrature can lead to an improvement in the convergence order of the iterative process. In this field, there are several available quadrature formulas, and for the purpose of our study, we have chosen to utilize the Trapezoidal formula, as defined by [2]

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)], \quad (2)$$

where

$$x_1 = x_0 + h.$$

In this paper, we will use another quadrature formula that is Simpson's $\frac{1}{3}$rd formula given by [2]

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)], \quad (3)$$

where

$$x_2 = x_0 + 2h.$$

We have combined the average of the two quadrature formulas in our research. The new quadrature formula is utilized to develop the new algorithm. Our paper consists of five sections. The first section is the introduction. The second section is method and description, where we formulate and implement the new quadrature rule to create the new iterative formula. The third section is convergence analysis, where we estimate the convergence order of the new iterative technique. The last section is numerical computation, where we use several examples to test the effectiveness of the new algorithm. The final section is the conclusion.

Numerous scientists, scholars, and researchers [3–10] have made significant contributions to this field.

2 | Method and Description

From *Eq. (2)*

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)].$$

When considering the aforementioned formula, it can be expressed in the following manner:

$$\int_{x_n}^{x} f'(t)dt = \left(\frac{x - x_n}{2}\right)[f'(x_n) + f'(x)] = I_T. \quad (4)$$

Again from *Eq. (3)*

$$\int_{x_0}^{x_2} f(x)dx = \left(\frac{x_2 - x_0}{6}\right)[f(x_0) + 4f(x_1) + f(x_2)].$$

In light of the aforementioned rule, we have arrived at the following conclusion:

$$\int_{x_n}^{x} f'(t)dt = \left(\frac{x - x_n}{6}\right)\left[f'(x_n) + 4f'\left(\frac{x + x_n}{2}\right) + f'(x)\right] = I_S. \quad (5)$$

In order to maintain clarity and accuracy, it is important to average the formulas defined in *Eq. (4)* and *Eq. (5)*.

$$\begin{aligned}\int_{x_n}^{x} f'(t)dt &= \frac{I_T + I_S}{2} \\ &= \frac{1}{2}\left[\left(\frac{x - x_n}{2}\right)\{f'(x_n) + f'(x)\} + \left(\frac{x - x_n}{6}\right)\left\{f'(x_n) + 4f'\left(\frac{x + x_n}{2}\right) + f'(x)\right\}\right] \\ &= \left(\frac{x - x_n}{3}\right)\left[f'(x_n) + f'\left(\frac{x_n + x}{2}\right) + f'(x)\right].\end{aligned} \tag{6}$$

Therefore

$$\int_{x_n}^{x} f'(t)dt = \left(\frac{x - x_n}{3}\right)\left[f'(x_n) + f'\left(\frac{x_n + x}{2}\right) + f'(x)\right].$$

From [3], we have

$$f(x) = f(x_n) + \int_{x_n}^{x} f'(t)dt.$$

From the above equation and *Eq. (6)* we get

$$0 = f(x_n) + \left(\frac{x - x_n}{3}\right)\left[f'(x_n) + f'\left(\frac{x_n + x}{2}\right) + f'(x)\right],$$

or

$$x = x_n - \frac{3f(x_n)}{\left[f'(x_n) + f'\left(\frac{x_n + x}{2}\right) + f'(x)\right]}.$$

Fixed point iterative from of the above equation will be

$$x_{n+1} = x_n - \frac{3f(x_n)}{\left[f'(x_n) + f'\left(\frac{x_n + x_{n+1}^*}{2}\right) + f'(x_{n+1}^*)\right]}.$$

Now replacing Newton's method in place of x_{n+1}^* present in the right- hand side of the above equation we have

$$x_{n+1} = x_n - \frac{3f(x_n)}{\left[f'(x_n) + f'\left(\frac{x_n + y_n}{2}\right) + f'(y_n)\right]}, \tag{7}$$

where

$$y_n = x_{n+1}^* = x_n - \frac{f(x_n)}{f'(x_n)}.$$

This is a two-step iterative implicit method.

3 | Convergence Analysis

This section will delve into the convergence analysis of the newly introduced method.

Theorem 1. Assuming that the function $f: X \subset \Re \rightarrow \Re$ for an open interval X possesses a simple root $\delta \in X$, and given that it is a sufficiently differentiable function in close proximity, it can be inferred that the iterative method defined in *Eq. (7)* exhibits a third-order convergence.

$$e_{n+1} = e_n^3\left(c_2^2 + \frac{1}{4}c_3\right) + O(e_n^4),$$

where

$$c_i = \frac{f^{(i)}(\delta)}{i!\ f'(\delta)},$$

and

$$e_i = x_i - \delta,\ \ i = 1,2,3,\ldots.$$

Proof: by using Taylor expansions about δ , we can write

$$f(x_n) = \dot{f}'(\delta)[\ e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + O(e_n^4)], \tag{8}$$

and

$$f'(x_n) = f'(\delta)[\ 1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + O(e_n^4)], \tag{9}$$

where

$$c_i = \frac{f^{(i)}(\alpha)}{i!\ f'(\alpha)}, \text{for} \quad i = 2, 3, \ldots$$

By dividing *Eq. (8)* by *Eq. (9)*, we get

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2 e_n^2 + 2(c_2^2 - c_3) e_n^3 + O(e_n^4).$$

Since

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$

we get

$$y_n = \delta + c_2 e_n^2 + 2(c_3 - c_2^2)\ e_n^3 + O(e_n^4). \tag{10}$$

Also, by using Taylor expansion about δ we have

$$f'(y_n) = f'(\delta)[1 + 2c_2^2\ e_n^2 + 4c_2\ (c_3 - c_2^2)\ e_n^3 + O(e_n^4)]. \tag{11}$$

Let

$$l_n = \frac{x_n + y_n}{2} = \frac{(e_n + \alpha) + \delta + c_2\ e_n^2 + 2\ (c_3 - c_2^2)\ e_n^3 + O(e_n^4)}{2}$$
$$= \delta + \frac{1}{2} e_n + \frac{c_2}{2} e_n^2 + \ (c_3 - c_2^2)\ e_n^3 + O(e_n^4).$$

$$f'(l_n) = f'(\delta)[1 + 2c_2 e_n + 3c_3\ e_n^2 + 4c_4\ \ e_n^3 + O(e_n^4).$$
$$l_n - \delta = \frac{e_n}{2} + \frac{c_2}{2} e_n^2 + \ (c_3 - c_2^2)\ e_n^3 + O(e_n^4).$$
$$(l_n - \delta)^2 = \frac{e_n^2}{4} + 2\frac{e_n}{2}\frac{c_2}{2} e_n^2 + O(e_n^4).$$
$$(l_n - \delta)^2 = \frac{e_n^2}{4} + \frac{c_2}{2} e_n^3 + O(e_n^4).$$
$$(l_n - \delta)^3 = \frac{e_n^3}{8} + O(e_n^4).$$

$$f'(l_n) = f'(\delta)\left[1 + 2c_2\left\{\frac{e_n}{2} + \frac{c_2}{2}e_n^2 + (c_3 - c_2^2)e_n^3 + O(e_n^4)\right\} + 3c_3\left\{\frac{e_n^2}{4} + \frac{c_2}{2}e_n^3 + O(e_n^4)\right\}\right.$$
$$\left. + 4c_4\left\{\frac{e_n^3}{8} + O(e_n^4)\right\}\right]$$
$$= f'(\delta)\left[1 + c_2e_n + e_n^2\left\{c_2^2 + \frac{3c_3}{4}\right\} + e_n^3\left\{2c_2(c_3 - c_2^2) + 3c_3\frac{c_2}{2} + \frac{c_4}{2}\right\}\right.$$
$$\left. + O(e_n^4)\right]$$
$$= f'(\delta)\left[1 + c_2e_n + e_n^2\left(c_2^2 + \frac{3c_3}{4}\right) + e_n^3\left\{2c_2c_3 - 2c_2^3 + \frac{3}{2}c_3c_2 + \frac{c_4}{2}\right\}\right.$$
$$\left. + O(e_n^4)\right]$$
$$= f'(\delta)\left[1 + c_2e_n + e_n^2\left(c_2^2 + \frac{3c_3}{4}\right) + e_n^3\left\{\frac{7}{2}c_2c_3 - 2c_2^3 + \frac{c_4}{2}\right\} + O(e_n^4)\right].$$

Assume

$$H_n = f'(x_n) + f'(l_n) + f'(y_n)$$
$$= f'(\delta)[\{1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + O(e_n^4)\}]$$
$$+ \left\{1 + 2c_2e_n + \left(c_2^2 - \frac{3}{4}c_3\right)e_n^2 + \left(\frac{7}{2}c_2c_3 - 2c_2^3 + \frac{c_4}{2}\right)e_n^3 + O(e_n^4)\right\}$$
$$+ \{1 + 2c_2^2e_n^2 + 4c_2(c_3 - c_2^2)e_n^3 + O(e_n^4)]$$
$$= f'(\delta)[3 + 3c_2e_n + e_n^2\{3c_3e_n^3 + c_2^2 + \frac{3}{4}c_3 + 2c_2^2\} + e_n^3\{4c_4 + \frac{7}{2}c_2c_3 - 2c_2^3$$
$$+ \frac{c_4}{2} + 4c_2c_3 - 4c_2^3)\} + O(e_n^4)]$$
$$= f'(\delta)[3 + 3c_2e_n + e_n^2(\frac{15}{4}c_3 + 3c_2^2) + e_n^3(\frac{9}{2}c_4 + \frac{15}{2}c_2c_3 - 6c_2^3)$$
$$+ O(e_n^4)].$$
$$\frac{3f(x_n)}{H_n} = \frac{e_n + c_2e_n^2 + c_3e_n^3 + O(e_n^4)}{\left[1 + c_2e_n + \left(\frac{5}{4}c_3 - c_2^2\right)e_n^2 + O(e_n^3)\right]}.$$

Taking

$$u = c_2e_n + \left(\frac{5}{4}c_3 - c_2^2\right)e_n^2 + O(e_n^3).$$

$$u^2 = c_2^2e_n^2 + 2c_2\left(\frac{5}{4}c_3 + c_2^2\right)e_n^3 + O(e_n^4).$$

$$u^3 = c_2^3e_n^3 + O(e_n^4).$$

$$\Big(1 + u)^{-1} = 1 - u + u^2 - u^3 + \cdots$$
$$= 1 - \left\{c_2e_n + e_n^2\left(\frac{5}{4}c_3 + c_2^2\right) + O(e_n^3)\right\} + \left\{c_2^2e_n^3 + 2c_2\left(\frac{5}{4}c_3 + c_2^2\right)e_n^3\right\}$$
$$- \{c_2^3e_n^3 + O(e_n^3)\} = 1 - c_2e_n + e_n^2\left\{\frac{-5}{4}c_3 - c_2^2 + c_2^2\Big)\right\} + O(e_n^3)$$
$$= 1 - c_2e_n - \frac{5}{4}c_3e_n^2 + O(e_n^3).$$

$$\frac{3f(x_n)}{H_n} = [e_n + c_2e_n^2 + c_3e_n^3 + \cdots] \times \left[1 - c_2e_n + \frac{5}{4}c_3e_n^2 + O(e_n^3)\right]$$
$$= e_n + c_2e_n^2 - \frac{5}{4}c_3e_n^3 + O(e_n^4) + c_2e_n^2 - c_2^2e_n^3 + O(e_n^4) + c_3e_n^3$$
$$= e_n + e_n^3\left(\frac{-5}{4}c_3 - c_2^2 + c_3\right) + O(e_n^4) = e_n + e_n^3\left(\frac{-1}{4}c_3 - c_2^2\right) + O(e_n^4).$$

$$x_{n+1} = e_n + \delta - \left\{e_n + e_n^3\left(-c_2^2 - \frac{1}{4}c_3\right) + O(e_n^4)\right\}.$$

We obtain the following error

$$e_{n+1} = e_n^3\left(c_2^2 + \frac{1}{4}c_3\right) + O(e_n^4).$$

This indicates that the iteration methods outlined in *Eq. (7)* demonstrate third-order convergence.

4 | Numerical Computation

We are pleased to present several examples to illustrate the effectiveness of the new technique. In our demonstration, we will compare Newton's Method (NM) with the process outlined in *Eq. (7)* (ALG1). We will be utilizing a specified tolerance$\epsilon = 10^{-15}$. The approximate roots provided are accurate up to 15 decimal places. The computations have been conducted using Matlab. Additionally, *Table 1* showcases the number of iterations (IT) required by each method to ascertain the root of the nonlinear equations.

Table 1. The number of iterations (IT) required by each method.

$f(x)$	x_0	Root δ	Number of Iteration (IT) NM	ALG1
$f_1(x) = x^3 + 4x^2 - 10$	0.3	1.365230013414096845823988159	8	5
$f_2(x) = \sin(x)^2 - x^2 + 1$	1.0	1.404491648215341226035086891	6	4
$f_3(x) = x^2 - e^x - 3x + 2$	2.0	0.257530285439860760455367304 7	5	4
$f_4(x) = \cos(x) - x$	1.7	0.739085133215160648637940089	4	3
$f_5(x) = (x-1)^3 - 1$	3.5	2.0	7	5
$f_6(x) = x^3 - 10$	1.5	2.154434690031883721759293565	div	div
$f_7(x) = 3x + \sin x - e^x$	0.8	0.360421702960324401369329515 8	5	3

Note: div strands for divergent.

The following graph represents the performance of methods NM and ALG1 in the sense of number of iterations.

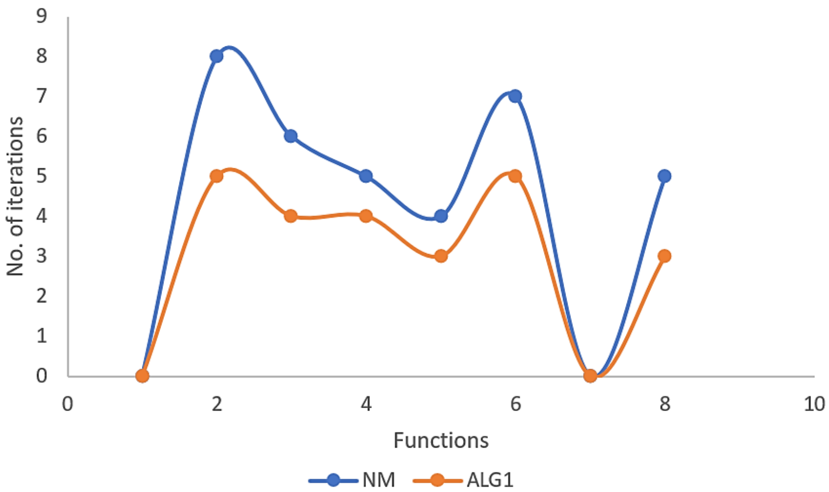

Fig. 1. The performance of methods NM and ALG1.

6 | Conclusion

In this study, a third-order iterative method was introduced for the purpose of solving nonlinear algebraic and transcendental equations with specified initial guesses. This method was developed through the amalgamation of the Trapezoidal and Simpson's formulas. The newly devised iterative scheme offers the roots of the equations in a reduced number of iterations compared to the previous method. Additionally, several examples were examined to assess the efficacy of the new method in accurately determining the roots of the equations up to 15 decimal places.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

All data are included in the text.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Gautschi, W. (2011). *Numerical analysis*. Birkhäuser Boston, MA. https://doi.org/10.1007/978-0-8176-8259-0

[2] Burden, R. L. & Faires, J. D. (2011). *Numerical Analysis*. Cengage Learning. https://www.amazon.com/Numerical-Analysis-Richard-L-Burden/dp/0538733519

[3] Weerakoon, S., & Fernando, T. (2000). A variant of Newton's method with accelerated third-order convergence. *Applied mathematics letters*, *13*(8), 87–93.

[4] Zhanlav, T., & Otgondorj, K. (2024). High efficient iterative methods with scalar parameter coefficients for systems of nonlinear equations. *Journal of mathematical sciences (United States)*, *279*(6), 866–875. DOI:10.1007/s10958-024-07066-4

[5] Shams, M., Kausar, N., Agarwal, P., & Oros, G. I. (2022). Efficient iterative scheme for solving non-linear equations with engineering applications. *Applied mathematics in science and engineering*, *30*(1), 708–735. DOI:10.1080/27690911.2022.2130914

[6] Khdhr, F. W., Saeed, R. K., & Soleymani, F. (2019). Improving the computational efficiency of a variant of Steffensen's method for nonlinear equations. *Mathematics*, *7*(3), 306. DOI:10.3390/math7030306

[7] Abdul-Hassan, N. Y., Ali, A. H., & Park, C. (2022). A new fifth-order iterative method free from second derivative for solving nonlinear equations. *Journal of applied mathematics and computing*, *68*(5), 2877–2886. DOI:10.1007/s12190-021-01647-1

[8] Saeed, R. K., & Aziz, K. M. (2013). Iterative methods for solving nonlinear equations by using quadratic spline function. *Mathematical sciences letters*, *2*(1), 37–43. DOI:10.12785/msl/020105

[9] Singh, T., & Behera, D. K. (2022). A new approach of iterative methods for solving non-linear equations using Boole's quadrature. *South east asian journal of mathematics and mathematical sciences*, *18*(1), 277–286.

[10] Bahgat, M. S. M. (2021). Three-point iterative algorithm in the absence of the derivative for solving nonlinear equations and their basins of attraction. *Journal of the egyptian mathematical society*, *29*(1), 1–17. DOI:10.1186/s42787-021-00132-9